////-----------------------------------------------------------------

//// 8) urunAdi=“Utu” olan veya vergi=10 olan dokumanlari listeleyiniz

////-----------------------------------------------------------------

db.fiyatlar.find({$or:[{"urunAdi": "Utu"},{"vergi":10}]});

////-----------------------------------------------------------------

//// 9) Vergi numarası 10 olmayan dokumanlari listeleyiniz

////-----------------------------------------------------------------

db.fiyatlar.find({"vergi":{$not: {$eq:10}}});

////-----------------------------------------------------------------

//// 10) urunAdi= TV ve vergi=10 olmayan dokumanlari listeleyiniz

db.fiyatlar.find( {$nor: [{"urunAdi":"TV"},{"vergi":10}]});

////-----------------------------------------------------------------

//// 11) urunAdi=TV olan kayitlarin sadece fiyat’ini listeleyiniz.

////     (Bu isleme Projection denilmektedir)

//// fiyatı göster id gösterme.

//// 1= true göster 0= false gösterme

//// SYNTAŞ = db.collection.find ( {filter},{projection})

////-----------------------------------------------------------------

db.fiyatlar.find({"urunAdi":"TV"},{"fiyat":1,"\_id":0});

// 12- Listesdeki tüm dokumanların yalnızca ürünadı ve fiyatı göster

db.fiyatlar.find({},{"urunAdi":1,"fiyat":1,"\_id":0});

////-----------------------------------------------------------------

//// 13) urunAdi=Buzdolabı olan kayitlarin sadece urunAdi ve fiyatini listeleyiniz.

////-----------------------------------------------------------------

db.fiyatlar.find({"urunAdi":"Buzdolabi"}, {"udurAdi":1, "fiyat":1, "\_id":0})

////=================================================================

////                    COMPARAISON OPERATOR

//// $eq : esit mi

//// $lt : kücük mü

//// $gt : buyuk mu

//// $lte: kucuk esit mi

//// $gte: buyuk esit mi

//// $ne : esit degil mi

//// $in : belirtilen degerler, bir dizideki degerlerden birisine esit mi

//// $nin: belirtilen degerler, bir dizideki tum degerlere esit degil mi

//// SYNTAX :   {field: {$gte: value} }

////=================================================================

////-----------------------------------------------------------------

//// 13) fiyatı 5000 olan urunleri listeleyiniz

////-----------------------------------------------------------------

db.fiyatlar.find({"fiyat":{$eq:5000}});

//db.fiyatlar.find({})

////-----------------------------------------------------------------

//// 14) fiyatı 3500'e esit veya daha fazla olan urunleri listeleyiniz

////-----------------------------------------------------------------

db.fiyatlar.find({"fiyat": {$gte:3500}});

// Sadece ürün adı ve fiyatını göster

db.fiyatlar.find({"fiyat": {$gte:3500}}, {"urunAdi":1,"fiyat":1, "\_id":0});

////-----------------------------------------------------------------

//// 15) fiyatı 4000'e esit veya daha kucuk olan urunleri listeleyiniz

////-----------------------------------------------------------------

db.fiyatlar.find({"fiyat":{$lte:4000}},{"urunAdi":1,"\_id":0});

////-----------------------------------------------------------------

//// 16) fiyatı 1500, 3500 veya 7000 olan urunleri listeleyiniz

////-----------------------------------------------------------------

db.fiyatlar.find({"fiyat":{$in:[1500,3500,7000]}},{"\_id":0});

//// Aynı soruyu or ile yapalmı.

////db.fiyatlar.find({$or:[{"fiyat":1500},{"fiyat":3500},{"fiyat":7000}]});

////-----------------------------------------------------------------

//// 17) fiyatı 1500, 3500 ve 7000 olmayan urunleri listeleyiniz

////-----------------------------------------------------------------

db.fiyatlar.find({"fiyat":{$nin:[1500,3500,7000]}},{"\_id":0});

////-----------------------------------------------------------------

//// 18) fiyatı 5000'den kucuk olup urunAdi Utu olmayan urunlern

//// SADECE urunAdi ve fiyat bilgilerini listeleyiniz

////-----------------------------------------------------------------

db.fiyatlar.find({"fiyat":{$lt:7000}, "urunAdi":{$ne:"Utu"}}, {"urunAdi":1, "fiyat":1, \_id:0});

////-----------------------------------------------------------------

//// 18) fiyatı 5000'den kucuk olan veya  urunAdi Koltuk olan urunlern

//// SADECE urunAdi ve fiyat bilgilerini listeleyiniz

////-----------------------------------------------------------------

db.fiyatlar.find({$or: [{"fiyat":{$lt:5000}},{"urunAdi":"Koltuk"}]},{"urunAdi":1,"fiyat":1,"\_id":0});

////-----------------------------------------------------------------

//// 19) urunlerin tamamini fiyatlarina gore artan sirada listeleyiniz.

////-----------------------------------------------------------------

db.fiyatlar.find().sort({"fiyat":1});

// 1= küçükten büyüğe  -1= Büyükten küçüğe sırala

////-----------------------------------------------------------------

//// 19) urunlerin tamamini fiyatlarina gore artan sirada listeleyiniz.

//   yalnızca urun adını göster, id gösterme. azalan sıraya göre göster

////-----------------------------------------------------------------

db.fiyatlar.find({},{"urunAdi":1, "\_id":0}).sort({"fiyat":-1});

////-----------------------------------------------------------------

//// 21) fiyati 5000 olan urunleri, urunAdi’na gore siralayarak (A-Z)

//// listeleyiniz.

// 1= A-Z   => -1 Z-Z  Not: Askii formatına göre sıralama yapar

////-----------------------------------------------------------------

db.fiyatlar.find({"fiyat":5000},{"\_id":0}).sort({"urunAdi":1});

// SYNTAX: db.cloneCollection.find (query,projection)

////=================================================================  
////            findOneAndUpdate - findOneAndReplace  
////=================================================================

// A - findOneAndReplace()   
//----------------------------  
// 1-) belirtilen koşullara uyan ilk sorguyu bulur ve degistirir.   
// 2-) Komut icerisinde belirtilen kismilari guncellerken bos birakilan   
// alanlari kaldirir. (API'lerdeki PUT metoduna benzetilebilir).  
// 3-) Islem sonunda VSCODE ilgili dokumanin guncellenmemiş halini gosterir.  
// 4-) SYNTAX  
// db.collection.findOneAndReplace( filter, replacement, options ) ;   
//   
// B - findOneAndUpdate()   
//----------------------------  
/// 1-) Belirtilen koşullara uyan ilk sorguyu bulur ve degistirir.   
// 2-) Komut icerisinde belirtilen kismilari guncellerken bos birakilan   
// alanlari modifiye etmez (API'lerdeki PACTH metoduna benzetilebilir).  
// 3-) komutun kosul kismindan sonra degislikileri gerceklestirmek icin bir   
// atomic operator kullanilir. ($set, $inc, $ mul vb.)  
// 4) Islem sonunda VSCODE ilgili dokumanin guncellenmemiş halini gosterir.  
// 5-) SYNTAX  
// db.collection.findOneAndUpdate( filter, update, options )¶  
////=================================================================

use("kisiler");

db.createCollection("personel");

db.personel.insertMany([

 {"isim":"Ali Bak", "maas": 5000, "iseBaslama": new Date('2014-03-01')},

 {"isim":"Can Tak", "maas": 7500, "iseBaslama": new Date('2011-03-04')},

 {"isim":"Veli Gel", "maas": 3550, "iseBaslama": new Date('2013-03-01')},

 {"isim":"Can Yak", "maas": 4800, "iseBaslama": new Date('2011-09-30')},

 {"isim":"Veli Git", "maas": 7200, "iseBaslama": new Date('2007-12-06')},

 {"isim":"Ali Han", "maas": 5250, "iseBaslama": new Date('2004-11-21')},

 {"isim":"Ayse Gel", "maas": 10500, "iseBaslama": new Date('1990-07-23')},

 {"isim":"Su Ak", "maas": 3300, "iseBaslama": new Date('1999-03-30')},

  ]);

////-----------------------------------------------------------------

//// 3) Maasi 4000 ’den az olan ilk personelin maasini 4000 TL yapiniz.

////-----------------------------------------------------------------

db.personel.findOneAndReplace({"maas":{$lt:4000}},{"maas":4000});

////-----------------------------------------------------------------

//// 4) Maasi 4800 olan personelin maasini 5000, isimni "Ali Can" ve

////   ise baslama tarhini su an yapalim.

////-----------------------------------------------------------------

db.personel.findOneAndReplace({"maas": 5250}, {"maas":5555, "isim":"Ceylan", "iseBaslama": new Date()} );